

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**  
**Факультет физико-математических и естественных наук**  
**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**  
**ЛАБОРАТОРНОЙ РАБОТЕ №4**  
*дисциплина: Архитектура компьютера*

Студент: Халилова Фатима Расуловна

Группа: НММбд-04-24

**МОСКВА**  
2024 г.

## **Содержание:**

1. Цель работы
2. Ход выполнения лабораторной работы
3. выполнение самостоятельной работы
4. Вывод
5. Список литературы

## **Цель работы**

Целью данной лабораторной работы является освоение процедуры компиляции сборки программ, написанных на ассемблере NASM

## **Ход выполнения лабораторной работы**

### **4.3.1. Программа Hello World!**

1. Я создала каталог для работы с программами на языке ассемблера NASM:

```
fkhalilova@fkhalilova-VirtualBox:~$ mkdir -p ~/work/arch-pc/lab04
```

2. Далее перешла в созданный каталог, создала текстовый файл с именем

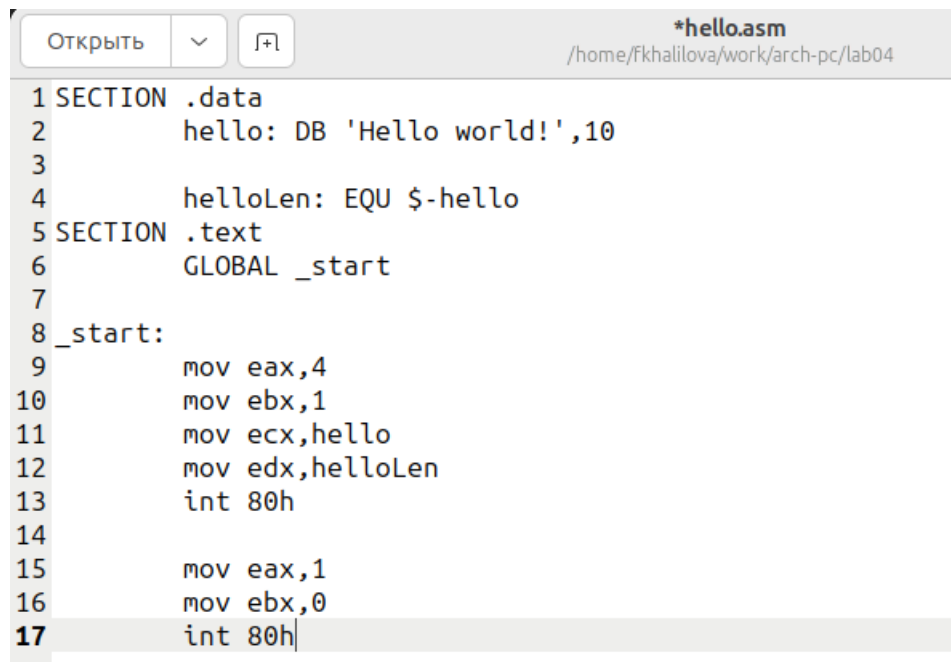
hello.asm

```
fkhalilova@fkhalilova-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab03/report$ cd
fkhalilova@fkhalilova-VirtualBox:~$ cd ~/work/arch-pc/lab04
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ touch hello.asm
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ gedit hello.asm
Gtk-Message: 14:38:34.962: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please try to
not load it.
```

3. открыла этот файл с помощью текстового редактора gedit:

```
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ gedit hello.asm
```

5. Далее я ввела в него следующий текст:



```
*hello.asm
/home/fkhalilova/work/arch-pc/lab04

1 SECTION .data
2     hello: DB 'Hello world!',10
3
4     helloLen: EQU $-hello
5 SECTION .text
6     GLOBAL _start
7
8 _start:
9     mov eax,4
10    mov ebx,1
11    mov ecx,hello
12    mov edx,helloLen
13    int 80h
14
15    mov eax,1
16    mov ebx,0
17    int 80h
```

### 4.3.2. Транслятор NASM

1. Я написала необходимую программу для компляции текста «Hello World!», текст программы был набран без ошибок, поэтому я проверила наличие созданного файла:

```
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list.lst  main  obj.o
```

3. Написала следующую команду и проверила ее наличие:

```
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list.lst  main  obj.o
```

### 4.4. Компоновщик LD

1. Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику, а для этого я написала соответствующую команду:

```
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

2. Я выполнила следующую команду:

```
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

Имя исполняемого файла - “obj.o”. Имя объектного файла – main.

#### 4.4.1. Запуск исполняемого файла

Я запустила на выполнение исполняемый файл. В результате в терминале вывелось “Hello World!”:

```
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ ./hello
Hello world!
```

### Выполнение самостоятельной работы:

1. В каталоге ~/work/arch-pc/lab04 с помощью команды `cp` я создала копию файла “hello.asm” с именем “lab4.asm”. Проверила наличие созданного файла:

```
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
```

2. С помощью текстового редактора gedit я внесла изменения в текст программы в файле “lab4.asm” так, чтобы вместо Hello world! на экран выводилась строка с моей фамилией и моим именем:

```
1 SECTION .data
2     lab4: DB 'халилова фатима',10
3
4     labLen: EQU $-lab4
5 SECTION .text
6     GLOBAL _start
7
8 _start:
9     mov eax,4
10    mov ebx,1
11    mov ecx,lab4
12    mov edx,lab4Len
13    int 80h
14
15    mov eax,1
16    mov ebx,0
17    int 80h
```

3. Я

оттранслировала полученный текст программы “lab4.asm” в объектный файл.

Далее выполнила компоновку объектного файла и запустила получившийся исполняемый файл:

```
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
```

```
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
fkhalilova@fkhalilova-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
```

4. Я

скопировала файлы “hello.asm” и “lab4.asm” в мой локальный репозиторий в каталог “~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/“:



5. И загрузила файлы на GitHub:

```
fkhalilova@fkhalilova-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
fkhalilova@fkhalilova-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master 22a5c79] feat(main): make course structure
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
fkhalilova@fkhalilova-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 287 байтов | 287.00 КиБ/с, готово.
Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:frkhalilova/study_2023-2024_arh--pc.git
9336c61..22a5c79 master -> master
```

## **Вывод:**

В ходе данной лабораторной работы я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.





## Список литературы:

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. *Newham C.* Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. *Robbins A.* Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. *Zarrelli G.* Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.

9. *Колдаев В. Д., Лупин С. А.* Архитектура ЭВМ. — М. : Форум, 2018.
10. *Куляс О. Л., Никитин К. А.* Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. *Новожилов О. П.* Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. *Робачевский А., Немнюгин С., Стесик О.* Операционная система UNIX. — 2-е изд. — БХВ-Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. *Столяров А.* Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: [http://www.stolyarov.info/books/asm\\_unix](http://www.stolyarov.info/books/asm_unix).
15. *Таненбаум Э.* Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. *Таненбаум Э., Бос Х.* Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).